UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/998,511	11/30/2001	Jeremy Alan Arnold	IBM / 194	6560

26517 7590 05/04/2005

WOOD, HERRON & EVANS, L.L.P. (IBM)
2700 CAREW TOWER
441 VINE STREET
CINCINNATI, OH 45202

EXAMINER	
PHAM, CHRYSTINE	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 05/04/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/998,511	ARNOLD ET AL.	
	Examiner	Art Unit	
	Chrystine Pham	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 November 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7-15, 17-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7-15, 17-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Amendment filed on November 29th 2004. The Applicants have canceled claims 6, and 16. Claims 1, 15, 18, 26-28, and 30 have been amended. Claims 1-5, 7-15, 17-30 are presented for examination.

Response to Arguments

2. Applicants' arguments filed November 29th 2004 have been fully considered but they are not persuasive.

The Applicants essentially contend that Nishimura (US 5845125) fails to teach "setting an inheritance breakpoint that is associated with a first program entity in the object-oriented computer program in which is identified a method in response to user input, and halting execution of the program during debugging in response to reaching an implementation of the method defined in a second program entity in the program that is different from the first program entity". The Applicants essentially contend that "Nishimura is directed setting breakpoints on individual objects, or instances, of a class, **rather than upon all instances of a class**", Nishimura, using the "object-type" breakpoints, "discloses the opposite configuration to that recited in claim 1". It is respectfully submitted that, the teaching of Nishimura is not limited to using the "object-type" breakpoints as alleged. On the contrary, in the Background Of The Invention, Nishimura discloses setting inheritance breakpoint in a method of a base class (i.e., first program entity) (see at least *base class, class [array], class [array] operation* col.2:8-col.4:40), and halting execution of the program in response to reaching an implementation of the method (see at least *execution, object "a", class [array] operation, indirect base classes, descendant class* col.4:9-40) defined in a class (i.e., second program entity that is different from the first program entity) deriving from the base class (see at least *derived class, class [character-string]* col.2:8-col.4:40). Furthermore, in the Detailed Description, Nishimura also discloses setting a inheritance breakpoint in a method of a base class (i.e., first program entity) (see at least *User Breakpoint*

Setting, address-type breakpoint, applied to all objects, inherited member function col.14:55-col.15:25), and halting execution of the program in response to reaching an implementation of the method defined in a class (i.e., second program entity that is different from the first program entity) deriving from the base class (see at least *User Breakpoint Setting, address-type breakpoint, applied to all objects, execution, control, user-specified location, inherited member function* col.14:55-col.15:25). Thus, not only does Nishimura teach setting breakpoints on an object basis, Nishimura discloses setting inheritance breakpoints on methods of class and its base class(es) so that execution is halted when methods are invoked on all objects of the class.

As per claim 14, in response to Applicants' argument that Coplien (US 5093914) falls short of disclosing the claim invention, that is to say, "halting execution of a program upon reaching an implementation of a method identified in an interface with which a breakpoint is associated", the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981). Furthermore, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In view of the forgoing discussion, the examiner considers claim rejections under USC 102(b) and 103(a) to be proper and maintained.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 3-4, 7-10, 13, 15-18, 20, 22-24, 26-30 are rejected under 35 U.S.C. 102(b) as being anticipated by Nishimura et al. (US 5845125), hereinafter, *Nishimura et al.*.

As per claim 1, *Nishimura et al.* teach an apparatus (e.g., see FIG.1 & associated text), a computer-implemented method (e.g., col.1:9-13), a program product stored in a signal bearing medium including transmission medium and recordable medium (e.g., col.9:15-22 & 28-32) for debugging an object-oriented computer program which is resident in a memory (e.g., col.2:3-7, see *source code storage section 7* FIG.1 & associated text), the method comprising:

(a) in response to user input (e.g., see *User Breakpoint Setting, user, breakpoint section 16, address-type breakpoint* col.14:54-col.15:25), setting an inheritance breakpoint that is associated with a first program entity in the object-oriented computer program and a method identified in the first program entity (see at least *base class, breakpoint, class [array], class [array] operation* col.2:8-col.4:40; see *address-type breakpoint, inherited member function* col. 15:1-5); and

(b) halting execution of the object-oriented computer program during debugging in response to reaching an implementation of the method defined in a second program entity in the object-oriented computer program that is different [and that depends] from the first program entity (see at least *execution, object "a", class [array] operation, indirect base classes, descendant class* col.4:9-40; *User Breakpoint Setting, address-type breakpoint, applied to all objects, execution, control, user-specified location, inherited member function* col.14:55-col.15:25).

As per claim 3, *Nishimura et al.* teach the computer-implemented method as applied to claim 1, wherein the first program entity is a first class that includes a second implementation of the method (e.g., *class [array] operation*, col.4:9-40), wherein the second program entity is a

second class/subclass that inherits from the first class (e.g., see *derived/child class, base class, class [character-string], class [array]*), and wherein the first implementation of the method in the second class overrides the second implementation of the method in the first class (e.g., *inheritance, operations, additional data, existing class* col.2:20-31).

As per claim 4, it recites limitations which have been addressed in claim 3, therefore, is rejected for the same reasons as cited in claim 3.

As per claim 7, *Nishimura et al.* teach the computer-implemented method of claim 6, wherein setting the inheritance breakpoint includes storing in a breakpoint data structure (e.g., see FIG.5 & associated text) an entry that identifies the first program entity and the method (e.g., see *object identification number & constructor* col.13:29-36, see FIG.11,12 & associated text, col.15:43-49).

As per claim 8, *Nishimura et al.* teach the computer-implemented method of claim 1, further comprising, during loading of a class in the object-oriented computer program, identifying each implementation of the method in the class and setting a breakpoint on such implementation (e.g., FIG.9 & associated text, col.4:21-33, col.15:11-14 & 23-26), wherein halting execution of the object-oriented computer program during debugging in response to reaching the implementation of the method includes reaching a breakpoint set on such implementation (e.g., see FIG.2 & associated text, col.4:34-40, col.16:4-7).

As per claim 9, it recites limitations which have been addressed in claim 8, therefore, is rejected for the same reasons as cited in claim 8.

As per claim 10, *Nishimura et al.* teach the computer-implemented method of claim 9, wherein setting a breakpoint on each implementation of the method includes setting a breakpoint

on a first statement in an implementation of the method (e.g., see FIG.30 & associated text, col.1:38-41).

As per claim 13, it recites limitations which have been addressed in claim 3, therefore, is rejected for the same reasons as cited in claim 3.

As per claim 15, *Nishimura et al.* teach a computer-implemented method (see at least) of debugging an object-oriented computer program (e.g., col.2:3-7, see *source code storage section 7* FIG.1 & associated text), the method comprising:

(a) receiving user input to halt program execution during debugging in response to reaching any of a plurality of implementations of a method in an object-oriented computer program (see at least *execution, object "a", class [array] operation, indirect base classes, descendant class* col.4:9-40; *User Breakpoint Setting, address-type breakpoint, applied to all objects, execution, control, user-specified location, inherited member function* col.14:55-col.15:25); and

(b) thereafter setting a breakpoint for at least a subset of the plurality of implementations such that execution of the object-oriented computer program will be halted in response to reaching any of the implementations on which a breakpoint has been set (see claim 1).

As per claims 17, 18, 20, they recite limitations which have been addressed in claims 8, 1, 3 respectively, therefore, are rejected for the same reasons as cited in claims 8, 1, 3.

As per claims 22-24, they recite limitations which have been addressed in claims 7, 9, 8 respectively, therefore, are rejected for the same reasons as cited in claims 7, 9, 8.

As per claims 26-30, they recite limitations which have been addressed in claims 1, 3, 8, and 15, therefore, are rejected for the same reasons as cited in claims 1, 3, 8, and 15.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 2, 5, 14, 19, 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Nishimura et al.* in further view of Coplien et al. (US 5093914), hereinafter, *Coplien et al.*.

As per claim 2, *Nishimura et al.* teach the computer-implemented method and apparatus as applied to claim 1 above. *Nishimura et al.* do not expressly disclose the first program entity as an interface that identifies the method, and wherein the second program entity is a class that implements the method. However, *Coplien et al.* disclose an apparatus (e.g., see FIG.1,2 & associated text) and a method for debugging an object-oriented computer program (e.g., see Abstract, see FIG.10,11 & associated text) comprising a first program entity which is an abstract class (e.g., col.6:26-35, col.8:54-56, col.8:62-col.9:17, col.9:34-45) or an interface that identifies the method (e.g., see *class Window* FIG.5 & associated text) and a second program entity which is a class that implements the method (e.g., see *class Xwindow, class SunviewWindow* FIG.5 & associated text, col.9: 34-45). *Coplien et al.* further disclose setting a breakpoint in a function of the first program entity and halting execution during debugging in response to reaching an implementation of the method defined in a second program entity that is different from the first program entity (e.g., col.8:45-53). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Coplien et al.* into that of *Nishimura et al.* to replace the first program entity with an object-oriented interface or abstract class which would produce the expected result with reasonable success. And the

motivation for doing so would have been that since interfaces and abstract classes are designed for inheritance which enables data/variables and methods from the base/superclass (i.e., abstract or interface) to be automatically inherited by the derived/subclass without having to define the same data/variables and methods in the subclass, thus minimizing the amount of coding required in implementing the subclass, and in the case of debugging the object-oriented classes, said inheritance feature eliminates the need to manually setting a breakpoint in all implementations (e.g., in the subclass) of the methods identified in the superclass.

As per claims 5, 14, 19, 21, they recite limitations which have been addressed in claim 2, therefore, are rejected for the same reasons as cited in claim 2.

7. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Nishimura et al.* as applied to claim 1 above, in further view of West (US 5740440), hereinafter, *West*.

As per claim 11, *Nishimura et al.* teach the computer-implemented method of claim 9. *Nishimura et al.* do not expressly disclose setting a breakpoint on a method call to an implementation of the method. However, *West* discloses a method of debugging an object-oriented program (e.g., see Abstract) wherein breakpoints are set on method calls (e.g., col.4:1-6). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *West* into that of *Nishimura et al.* to enable setting breakpoint on method calls with reasonable success. And the motivation for doing so would have been that setting breakpoints on a method calls enables the state of the program to be examined at the suspension of execution, and determination can be made as to whether an object in which the method call is contained is newly created so that updates can be reported to the debugger for data tracking or other debug operations.

8. Claims 12 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Nishimura et al.* as applied to claims 1 and 18 above, in further view of Applicant's Admission of Prior Art (hereinafter APA)(see page 2 – Background of the invention).

As per claim 12, *Nishimura et al.* teach the computer-implemented method of claim 1. *Nishimura et al.* do not expressly disclose setting the inheritance breakpoint includes associating a user-specified condition with the inheritance breakpoint, and wherein halting execution of the object-oriented computer program during debugging in response to reaching the implementation of the method is performed only if the user-specified condition has been met. However, APA discloses associating a user-specified condition with the breakpoint , and wherein halting execution of the object-oriented computer program during debugging in response to reaching the implementation of the method is performed only if the user-specified condition has been met (e.g., see *after the breakpoint has been hit X times* pg.2:13-18). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of APA into that of *Nishimura et al.* to obtain conditional (i.e., user-specified condition) breakpoints with reasonable success. And the motivation for doing so would have been determining and halting execution only when the user-specified condition (i.e., breakpoint has been hit X times) for a breakpoint has been met would allow the user of the debugging method more control over program suspension. That is to say, by specifying and monitoring said condition associated with breakpoint A, the user could skip over X-1 breakpoint iteration(s) (which entail X-1 suspensions of program execution) and jump to the breakpoint of interest (i.e., when breakpoint A has been hit X times), thus making program debugging a more effective and efficient process.

As per claim 25, it recites limitations which have been addressed in claim 12, therefore, is rejected for the same reasons as cited in claim 12.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

1. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
May 2, 2005

TUAN DAM
SUPERVISORY PATENT EXAMINER